An Integrated System to support Internet of Things (IoT) in a Smart Home

Software Architecture (Autumn 2015) – IT University of Copenhagen

Group Project Phase 1 Submission

Henrik Buch-Larsen

Michael Poulin Mortensen

Rasmus Løbner Christensen

Table of Content

[2 Introduction 4](#_Toc432951565)

[3 User Stories 5](#_Toc432951566)

[4 Quality Attribute Scenarios 6](#_Toc432951567)

[4.1 Availability 6](#_Toc432951568)

[4.2 Scalability 7](#_Toc432951569)

[4.3 Modifiability 8](#_Toc432951570)

[4.4 Performance 9](#_Toc432951571)

[4.5 Security 9](#_Toc432951572)

[5 IoT Architecture - IoT in a Smart Home 10](#_Toc432951573)

[5.1 Application 10](#_Toc432951574)

[5.2 Management 10](#_Toc432951575)

[5.3 Service Organization 10](#_Toc432951576)

[5.4 IoT Process Management 11](#_Toc432951577)

[5.5 Virtual Entity 11](#_Toc432951578)

[5.6 IoT Service 11](#_Toc432951579)

[5.7 Communication 11](#_Toc432951580)

[5.8 Security 11](#_Toc432951581)

[5.9 Device 12](#_Toc432951582)

[6 System Architecture description 12](#_Toc432951583)

[6.1 Context Diagram 12](#_Toc432951584)

[6.2 Management services ( and Monitoring) 13](#_Toc432951585)

[6.3 Iot Services 14](#_Toc432951586)

[6.4 Iot Process Management Services 15](#_Toc432951587)

[6.5 Virtual Entity Services 16](#_Toc432951588)

[6.6 Devices 17](#_Toc432951589)

[6.7 Device discovery 18](#_Toc432951590)

[6.8 Security 18](#_Toc432951591)

[7 Developing a High-Level Architecture 19](#_Toc432951592)

[7.1 Logical View 21](#_Toc432951593)

[7.2 Process View 21](#_Toc432951594)

[7.3 Physical View 21](#_Toc432951595)

[8 Architecture Decision Log 22](#_Toc432951596)

[8.1 Design decisions 22](#_Toc432951597)

[8.1.1 Availability 22](#_Toc432951598)

[8.1.2 Modifiability 23](#_Toc432951599)

[8.1.3 Scalability 25](#_Toc432951600)

[8.1.4 Performance 26](#_Toc432951601)

[8.1.5 Security 27](#_Toc432951602)

[9 References 29](#_Toc432951603)

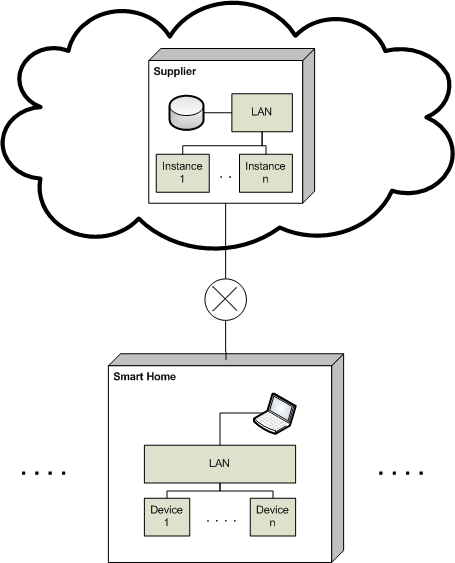
[9.1 Additional Links 29](#_Toc432951604)

[10 Notes 30](#_Toc432951605)

# Introduction

This document contains a proposed architecture for an integrated system to support Internet of Things (IoT) in a Smart Home. The proposed solution takes base in the IoT reference architecture, handed out as part of the course learning material.

Since the term Internet of Things in its nature is about collecting and exchanging data across a network, we believe that a distributed cloud solution is the best way of designing such a system. As an introduction, we present the below picture which describes the proposed system in a high level way.



For each Smart Home connected to the distributed solution, the given supplier of such a system stores an instance. Through the cloud every Smart Home can interact with the given services and processes which the system provides. The laptop in the Smart Home depicts the fact that the given household users do not require any supplier made equipment in order to interact with the system, simply a laptop (in general a computer with internet connection), in order to configure the system the first time accessing the solution, and furthermore handling the ongoing configuration that such a system requires.

# User Stories

At the initial launch meeting, the project team developed the following *User Stories*.

For all *User Stories* the structure of: As a <Role> I want <Action> So that <Concern/Benefit>.

Not all requirements in the assignment are described in this section.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Role | Action | Concern/Benefit |
| US0 | User | Restrict/Limit the use of Streaming/TV | So that children can watch and listen to music only at certain times. |
| US1 | User | Register a Device | It is useable. |
| US2 | User | Register Consumables | It is consumable by other User’s. |
| US3 | User | Register a Consumable | Orders can be made on specified stock thresholds. |
| US4 | User | Notify Fire Department, in case of fire | The Fire Department is aware, and takes the needed action. |
| US5 | User | Light up paths for an emergency exit, in case of fire | House residents know where to go in order to escape the house as fast as possible. |
| US6 | User | Override system security, in case of fire | No security measures are hindering an easy escape for the house residents. |

Additional brainstorm exercises were was done in this phase of the project, which can be seen under the ***notes*** section lastly in this document.

# Quality Attribute Scenarios

This section describes some of the relevant Quality Attribute Scenarios of the system.

The following tables describe the selected quality attributes and describe the motivation behind the choice.

|  |  |
| --- | --- |
| **Quality Attribute** | **Motivation** |
| Availability | Availability covers the system’s ability to survive situations like power or communications failure. The users of a Smart Home IoT system need to be guaranteed that the system acts correctly in the case of an emergency. |
| Scalability | A distributed cloud solution which should be able to manage and control potentially all the devices/sensors in all the households on the planet, face some pretty clear scalability challenges, which makes this quality attribute absolutely key. |
| Modifiability | A key function of the system is the ability to create and manage devices in the Smart Home. Furthermore the users should be able to manage e.g. food thresholds for the given household. Therefore we believe that modifiability is an important quality attribute of the proposed solution. |
| Performance | The performance of the system is also an important factor. The obvious example is in case of an emergency (e.g. fire), where the system *must* be able to take the correct action, as well with the desired response times. |
| Security | Security is the last quality attributed which we see necessary to include as relevant for the solution. The reason for choosing this is also quite obvious, since the system also acts as a security system for the given household as a whole. |

## Availability

A device connected to the system must be able to communicate with the system when a power failure occurs.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | Device |
| Stimulus | Service call to device or data logging from device |
| Environment | Power failure |
| Artifact | System |
| Response | A device must be able to route messages to the system when a power failure occurs. |
| Response Measure | System should be able to route service call within 1 seconds from sending system to device. |

A device connected to the system must be able to communicate with the system when the internet connection fails.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | Device |
| Stimulus | Service call to device or data logging from device |
| Environment | Internet failure |
| Artifact | System |
| Response | A device must be able to route messages to the system when a power failure occurs. |
| Response Measure | System should be able to route service call within 1 seconds from sending system to device. |

## Scalability

The system should be able to contain and manage at least 100 devices pr. customer instance.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | Device |
| Stimulus | Service call to device or data logging from device |
| Environment | Normal condition |
| Artifact | System |
| Response | System should route messages to and from devices and application without delay |
| Response Measure | System should be able to route service call within 1 seconds from sending system to device. |

The system should be able contain and manage concurrent devices pr. customer instance.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | Device |
| Stimulus | Concurrent calls to device or data logging from device |
| Environment | Normal condition |
| Artifact | System |
| Response | System should route messages to and from multiple devices and application without delay |
| Response Measure | System should be able to route concurrent service calls within 1 seconds from sending system to devices |

## Modifiability

The system should enable dynamically to allocate and remove devices to and from the house management system.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | User Interface |
| Stimulus | Register new device |
| Environment | Normal condition |
| Artifact | System |
| Response | System should be able to interact with new device without restart |
| Response Measure | New device should be immediately manageable |

The system should enable dynamically to allocate and remove Consumable to and from the house management system.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | User interface |
| Stimulus | Register new Consumable |
| Environment | Normal condition |
| Artifact | System |
| Response | System should be able to register new Consumables without restart |
| Response Measure | New Consumable should be immediately manageable |

The system should enable dynamically to setup threshold values for Consumables to and from the house management system.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | User interface |
| Stimulus | Setup threshold values for Consumable |
| Environment | Normal condition |
| Artifact | System |
| Response | System should be able to setup threshold values for Consumables without restart |
| Response Measure | Consumable threshold policy should be immediately in effect |

## Performance

The system must prioritize certain types of messages – e.g. fire alarms.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | System |
| Stimulus | A fire alarm call from a device |
| Environment | System is busy |
| Artifact | System |
| Response | The system must ensure that certain kinds of messages must be prioritized, so fire alarms are not delayed in the system. |
| Response Measure | System should be able to route fire alarm calls to external service providers within 5 seconds. |

## Security

The system should identify the user based on the device logging on to the Wi-Fi network.

|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | Security device |
| Stimulus | Device arrives at house Wi-Fi perimeter |
| Environment | Normal condition |
| Artifact | System |
| Response | User should be logged on the system |
| Response Measure | Request user pin code to verify identity |

The system should enable setting up difference user groups for the system.

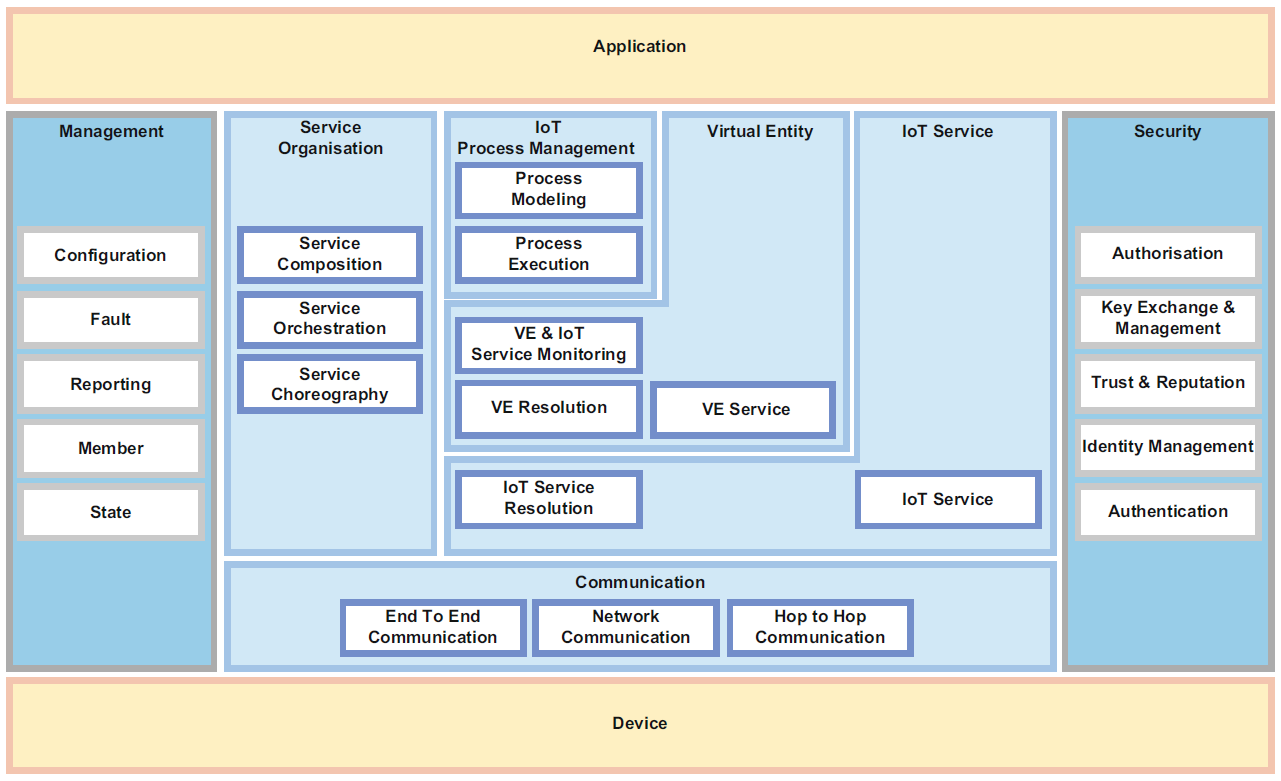
|  |  |
| --- | --- |
| **Elements** | **Refined General Scenario** |
| Source | Administration |
| Stimulus | User wants to setup the different User groups to apply for the system |
| Environment | Normal condition |
| Artifact | System |
| Response | New User groups have been enabled without restart |
| Response Measure | New User groups security levels have been defined |

# IoT Architecture - IoT in a Smart Home

The section serves the purpose of explaining which key parts from the IoT reference architecture that is being used in the solution which we offer. Furthermore we argue why certain parts of the original IoT reference architecture is left out.

For each of the below Groups a description will be given, containing either an explanation of the purpose of the Group (*in our proposed architecture*), or reason for being omitted in the overall solution.

This is done in order to give the reader a high-level understanding of how we actually used the IoT reference architecture. In the next section we present the actual system architecture description.



## Application

This is the top layer of the whole architecture - the application layer of the architecture.

## Management

Configuration of the system is handled through the embedded components in this Group, but also the ability for the system to keep track of the different Devices current configuration. Furthermore this Group is responsible for error handling in the system. If a given component (in the Architecture) is triggering an error, all messages should go through this Group in order to identify the nature and severity of the problem. Error logging should also be done at this location.

## Service Organization

This Group is not included in the proposed architecture due to the set of services being an already defined set. More precisely we see this Group as offering the possibility for the users to create and modify given services which the system offers, but in the proposed solution this is something that the users are not able to do. The reasoning behind this design choice is that we believe that handing out this functionality for the users, is something that would complicate the usage of the system to a level which is above the benefits which this may cause.

## IoT Process Management

All the different processes exposed by the system are modelled in this Group. This is where the process management is handled, e.g. what happens when a Fire alarm is triggered. Execution of processes is triggered here, and the piping of information (from e.g. a device) to the correct receivers as well.

## Virtual Entity

The Virtual Entity Group describes the actual entities in the system (e.g. devices). Furthermore entity specific services are also present in this Group. Registering and discovery of new devices, dispatching of relevant information between devices and getting the different device status is handled within this Group.

## IoT Service

The concrete services within the system reside in this Group. We’ve chosen to split the exposed service functionality into six distinct groups, in order to visualize the services which the system offers in a more structured way. These are the following:

* Alert services
* Order services
* Climate control services
* Pet feeding services
* House Control Access services
* Device control services

Each of these contains the desired functionality, related to the different areas of functionality which the system offers.

## Communication

This Group maintains communication between the different Groups in the IoT architecture. Furthermore this is where communication between the different devices is modelled. Later we present our Physical view of the communication layer, which goes more in-depth on how the actual communication is being handled. The devices are able to communication over Wi-Fi but also via 4G network, in case of the internet connection being unavailable.

## Security

This Group of functionality handles User authentication, and should prevent any tempering of data and/or information. It should also enable the possibility of creating User Groups, so that the users of the system are able to differentiate on the level of administration rights within the system, e.g. changing preferences stores within the system.

## Device

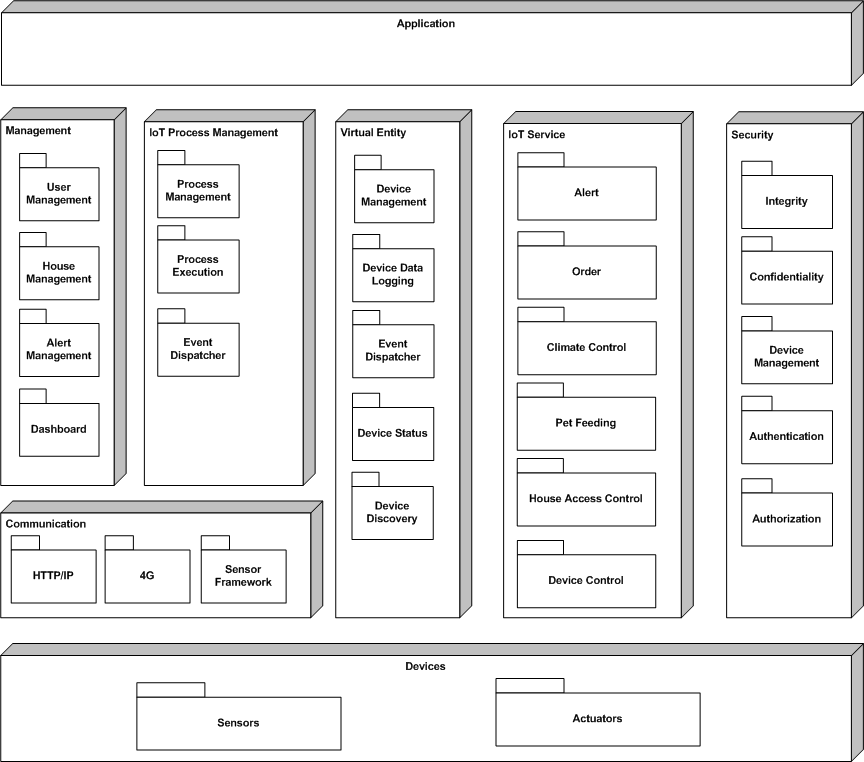
This Group is where the different devices are being handled. We’ve chosen to divide the actual physical devices into sub-groups, depending on role and behavior. In the following we present this division.

# System Architecture description

With the above in mind, we now present the system architecture description for the solution which we propose. For each of the above stated Groups, we present the related services which the architecture must enable. Responsibility and methods are also declared for each of the Groups.

First of all we present the context diagram (in relation to the above picture of the IoT reference architecture), which should give the reader an overview of our architecture.

## Context Diagram



## Management services ( and Monitoring)

|  |  |
| --- | --- |
| Service Name | User Management |
| Responsibility | Management of Users and Pets |
| Methods | * CreateUser * GetUser * UpdateUser * DeleteUser * CreatePet * GetPet * UpdatePet * DeletePet |
| Collaboration | House Management |

|  |  |
| --- | --- |
| Service Name | House Management |
| Responsibility | Management of House and Rooms |
| Methods | * CreateHouse * GetHouse * UpdateHouse * DeleteHouse * CreateRoom * UpdateRoom * DeleteRoom |
| Collaboration | User Management, Device Management |

|  |  |
| --- | --- |
| Service Name | Alert Management |
| Responsibility | Management of Alerts. Alerts are rules that describes who should be alerted in different situations. External and internal receivers are also managed. |
| Methods | * CreateAlert * GetAlert * UpdateAlert * DeleteAlert * CreateReceiver * AddReceiverToAlert * RemoveReceiverFromAlert * AddEventTypeToAlert * RemoveEventTypeFromAlert |
| Collaboration | User Management, Device Management |

|  |  |
| --- | --- |
| Service Name | Dashboard (not UI) |
| Responsibility | Listing relevant management and runtime information for use in applications like web applications or mobile apps |
| Methods | * GetAlerts * GetUserStatus * GetRessourceStatus * GetSensorStatus |
| Collaboration | User Management, Device Management ... |

## Iot Services

|  |  |
| --- | --- |
| Service Name | Alert |
| Responsibility | This service executes Alerts based on rules from alert configuration. |
| Methods | * Alert |
| Collaboration | Alert Management, Process Execution |

|  |  |
| --- | --- |
| Service Name | Order |
| Responsibility | This service executes an Order to a predefined Order service . It can order any type of products that are predefined in the product catalog. |
| Methods | * Order |
| Collaboration | Process Execution |

|  |  |
| --- | --- |
| Service Name | Climate Control |
| Responsibility | This service controls the devices that are used for climate regulation in the house. This includes aircondition, heat and light. |
| Methods | * Alert |
| Collaboration | Alert Management, Process Execution |

|  |  |
| --- | --- |
| Service Name | Pet Feeding |
| Responsibility | This service controls the pet food dispensers in the house. |
| Methods | * Feed(Pet) |
| Collaboration | Process Execution |

|  |  |
| --- | --- |
| Service Name | House Access Control |
| Responsibility | This service controls the devices that are used for accessing the house such as locks on doors and windows. |
| Methods | * Unlock(Door/Window) * Lock(Door/Window) * Status(Door/Window) |
| Collaboration | Dashboard service, Process Execution |

|  |  |
| --- | --- |
| Service Name | Device Control |
| Responsibility | This service controls the devices that are turned on or off such as coffee machines and tv’s. |
| Methods | * On(Device) * Off(Device) * Status(Device) |
| Collaboration | Dashboard service, Process Execution |

## Iot Process Management Services

This package contains services for process management and configuration, process execution and the specific process services

|  |  |
| --- | --- |
| Service Name | Process Management |
| Responsibility | This service manages and configures the predefined processes.  The Fire Process is executed when a fire is detected  The UnauthorizedAccess Process is executed when a UA is detected |
| Methods | * Enable(Process) * Disable(Process) * SetSchedule(Process) * FireProcessConfiguration(FireConfig) * UnauthorizedAccessProcessConfig(UAConfig) |
| Collaboration | Process Execution |

|  |  |
| --- | --- |
| Service Name | Process Execution |
| Responsibility | This service executes the specific activities in the processes |
| Methods | * Execute(Process) |
| Collaboration |  |

|  |  |
| --- | --- |
| Service Name | Event Dispatcher |
| Responsibility | This service receives and parses events from the devices in the system and starts process execution based on the process configurations.  If a fire event is received from a device in the system, the Fire Process is started |
| Methods | * EventHandler(Event) |
| Collaboration |  |

## Virtual Entity Services

|  |  |
| --- | --- |
| Service Name | Device Management |
| Responsibility | This service registers devices in the system |
| Methods | * CreateDevice * GetDevice * UpdateDevice * DeleteDevice |
| Collaboration |  |

|  |  |
| --- | --- |
| Service Name | Device Data Logging |
| Responsibility | This service collects and stores data from devices that are logging data into the system |
| Methods | * Logevent |
| Collaboration |  |

|  |  |
| --- | --- |
| Service Name | Event Dispatcher |
| Responsibility | This service dispatches events to relevant subscribers |
| Methods |  |
| Collaboration |  |

|  |  |
| --- | --- |
| Service Name | Device Discovery |
| Responsibility | This service is responsible for discovery of active devices in the network and make them available for management |
| Methods |  |
| Collaboration |  |

|  |  |
| --- | --- |
| Service Name | Device Status |
| Responsibility | This service is responsible for monitoring the status of devices in the system on a schedule. If a device is offline an alert should be generated. |
| Methods |  |
| Collaboration |  |

## Devices

The physical devices are divided into several groups depending on their role and behavior. All devices are constructed from two main types of components: Sensors and Actuators.

The following table shows the main device types that play a role in the smart house and the type of sensors and actuators needed to realize the functionality.

|  |  |  |  |
| --- | --- | --- | --- |
| **Device Type** | **Smart Home Functionality** | **Sensors** | **Actuator** |
| TV | On/Off on schedule. | Power Status (On/Off) | On/Off Power Switch |
| Coffee machine | On/Off on schedule. | Power Status (On/Off) | On/Off Power Switch |
| Refrigerator | Status on inventory | Inventory type selector  Inventory increase/decrease buttons | N/A |
| Pet Feeding Device | Feed Pet on schedule | Food in Dispenser Status.  (Yes/No) | On/Off Switch to control motor on dispenser |
| Fire Alarm | Alert in case of fire | Smoke Sensor (On/Off) | N/A |
| Door/Window Lock | Control Door Locks | Door Status Open or Closed (On/Off) | Switch to control motor on/off and direction that locks or unlocks door |
| Card Reader | Provide User Identity | Smart Card Reader(Cert)  Pin Code Pad (Pin) | N/A |
| Climate Control | Control aircondition and lights | Humidity and temperature sensors | Motor controlled termostats or aircondition devices  Motor controlled light regulators |
| Garage Door | Open or close the garage door when car arrives or leaves the home | Photoelectric sensor | Garage door motor |
| Movement Sensor | Detecting movement in a room |  |  |
| Rome Access Detection |  | Photoelectric sensor |  |

There are several standards emerging in the IoT domain that seeks to standardize the way these type of devices are connected to a larger system as the one being designed here.

As this assignment is not about these standards one that looks simple and appropriate for this case has been chosen as a base for further design activities.

## Device discovery

Device discovery is needed to establish link devices to the Smart Home System. There are several standards for this process. As this is not a primary topic of the assignment this will not be described in detail, but the system needs to have this type of functionality.

The main technologies in this area are “Bluetooth beacons”, “Wifi Aware” and “Physical Web”.

## Security

We need the following security services in the system:

*Integrity:*

The system must provide data integrity to ensure that altering of data is prevented.

TODO: Kryptering mellem hus-LAN og skyen.

*Confidentiality:*

Some of the information that are generated and processed by the system could be sensitive and therefore the system needs to provide data and message confidentiality.

TODO: beskriv hvordan vi hemmeligholder “stuff”. E.g. kamera billeder skal ikke på nettet.

*Authentication:*

The system should support secure authentication of the users. Both accesses to the system functionality and to the house itself should be based on proper authentication.

TODO: Hvordan authenticor vi brugeren I systemet!?!?

*Authorization:*

The system should provide possibility to setup user groups, so that certain users in the household can access functionality which other users cannot. E.g. changing food preferences for the household should only be done by certain users.

TODO: Beskriv noget om roller 🡪 brugere, administratorer etc.

# Developing a High-Level Architecture

The overall system architecture is distributed by nature as it includes many different devices that need to communicate with the main system and the stakeholders of the system.

The candidates for the main architectural style for the solution where ***broker*** and ***layered*** architectures.

*Broker:*

This pattern is used where processes and services typically are distributed across multiple nodes. The pattern provides location transparency to the service consumers and a centralized repository for service resolution.

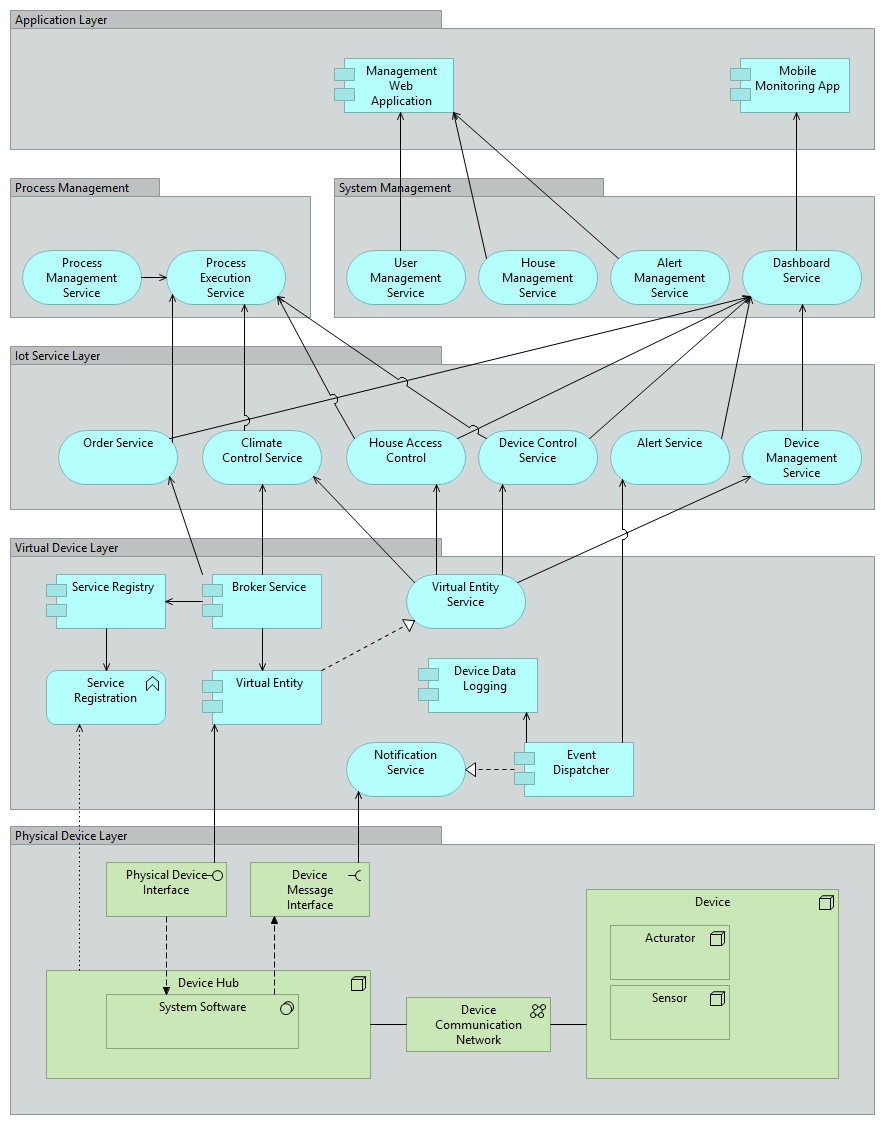
*Layered Architecture:*

The layered architecture is used to reduce complexity in a system. A layer typically includes related functionality such as resource access functionality, business logic or presentation.

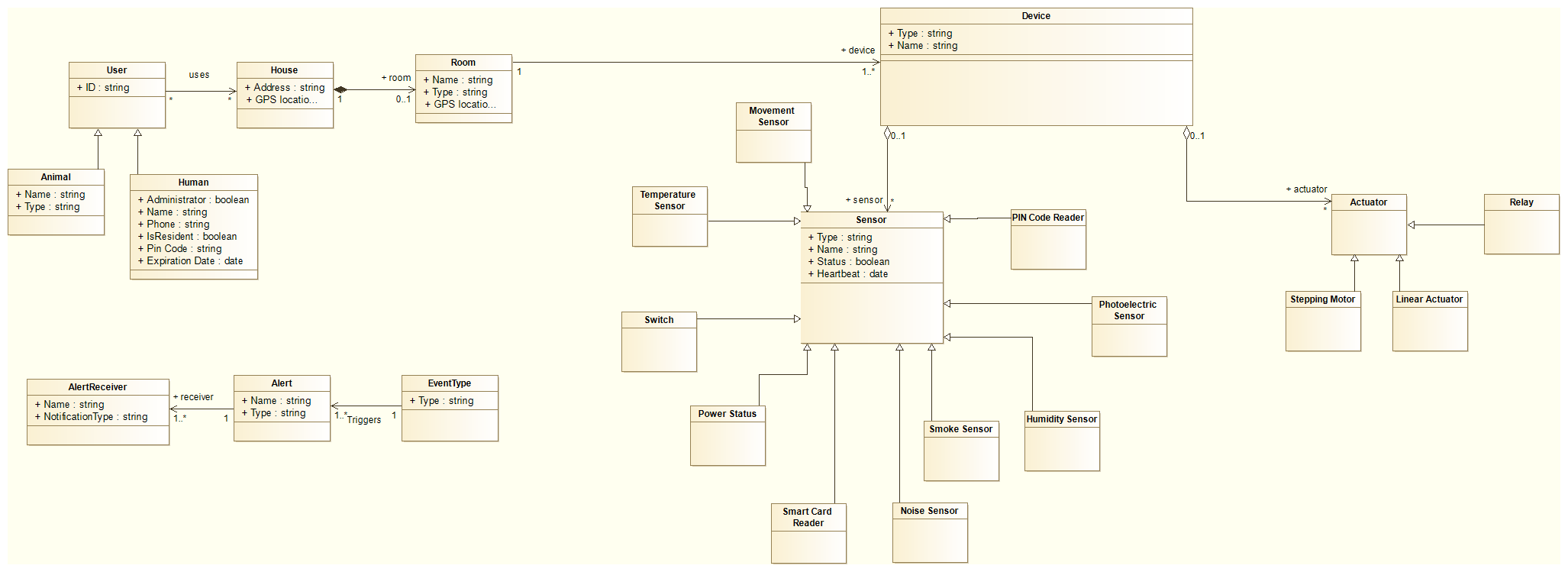
The final design of the Smart Home System applies the two architectural styles together in the following way:

The discovery and registration of devices and services uses the broker architecture to provide easy and uniform access to these services for the other components in the system. As the reference architecture is divided into several areas that also represent different levels of granularity, the layered architecture is combined with the broker style to provide the final system architecture.

The overall system architecture is presented as a figure in the following page.



## Logical View



## Process View

TODO: Sequence diagrams

## Physical View

TODO: Deployment

# Architecture Decision Log

This section contains descriptions of various design decisions we’ve made in order to identify the required services. The design decision descriptions will be divided into the five categories of quality attributes which we stated earlier in this report (Availability, Scalability, Modifiability, Performance & Security).

## Design decisions

### Availability

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Concern (Identifier: Description) | | T system must be able to communicate information about emergency alarms (e.g. fire alarm) even in the case of a power failure. |
|  | Ranking criteria (Identifier: Name) | | 1. Availability |
|  | Options | Identifier: Name | 1. In-house emergency routine storing |
|  | Description | A solution where the given Smart Home had some sort of device running (physical machinery, e.g. a dedicated computer), which could take over the handling of device information in case of power failure. This device should obviously be equipped with some sort of battery, in order to function without power. This way the different emergency processes would still be upheld by this device, even in case of power failure. |
|  | Status | The option is declined. |
|  | Relationship(s) | Alert handling |
|  | Evaluation | The evaluation is that this solution would require additional support from the supplier. The supplier would have to ensure that this machinery would always be functioning, as well as guarantee that the hardware was not malfunctioned. Imagine having the entire worlds households connected to the solution, this would require some support. |
|  | Rationale of decision | Since the system is thought out to be a complete cloud solution, which in theory should handle all the households in the world, this solution is simply not feasible. |
|  | Identifier: Name | 1. Cloud emergency routine storing |
|  | Description | A solution where all emergency routines are stored in the Cloud, like the rest of the system. If a power failure occurs, the current devices would still be functioning, since they all have a backup battery. The given device would in case of an emergency realize that an internet connection was no available, and then use 4G (mobile communication) instead. This way the system would receive the emergency readings from the sensor, and proceed by activation the correct emergency process. |
|  | Status | Option is decided. |
|  | Relationship(s) | Alert handling |
|  | Rationale of decision | This solution would require all emergency devices (the ones detecting e.g. a fire), to be equipped with a battery and a 4G SIM card. This way communication would always be ensured, even in the case of power failure. |

### Modifiability

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Concern (Identifier: Description) | | Should the users of the system be able to create their own processes? E.g. setting together some personal sequence of services which the system offers. For instance adding additional steps in one of the emergency routines. |
|  | Ranking criteria (Identifier: Name) | | 1. Modifiability |
|  | Options | Identifier: Name | Yes – the users should be able to create processes |
|  | Description | A solution where each user has the ability to create and modify the different processes exposed by the system. |
|  | Status | The option is declined. |
|  | Relationship(s) | Process management |
|  | Evaluation | *We believe that this would create a lot of uncertainty while using the processes. As a supplier you could no longer guarantee that your processes actually worked. Worst case scenario is imagining a fire emergency process which does not actually contact the correct Fire Department, due to the process being modified in an in-correct way.* |
|  | Rationale of decision | *The reasoning behind this design choice is that we believe that handing out this functionality for the users, is something that would complicate the usage of the system to a level which is above the benefits which this may cause.* |
|  | Identifier: Name | No – the system comes with predefined processes |
|  | Description | A solution where all the given processes which the system exposes, is non-modifiable. |
|  | Status | Option is decided. |
|  | Relationship(s) | Process management |
|  | Rationale of decision | Since we believe this would create a much more stable system, where users could actually trust the exposed processes, we have taken the decision of limiting modifiability in this regard. |

### Scalability

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Concern (Identifier: Description) | | The system must be able to handle a different numbers of devices from multiple users, and must be able to react on alarms without delays |
|  | Ranking criteria (Identifier: Name) | | 1. Scalability |
|  | Options | Identifier: Name | Own servers (Load balancing architecture) |
|  | Description | It is an option to have a server solution to run the system. The System Owner will have a better control over the system and know where the data is stored. The system could for instance be based on a load balancing architecture where new servers can be added when new customers are added to the system. |
|  | Status | This option is declined. |
|  | Relationship(s) |  |
|  | Evaluation | If using a load balancing architecture new servers must be added when needed. |
|  | Rationale of decision | This option is declined because it will require much work to keep the numbers of servers required if many there are many customers going to use the system. Furthermore, the number of devices for each customer can be as high as 100, so the impact of each new customer on the system can be hard to predict |
|  | Identifier: Name | Cloud solution |
|  | Description | It is an option to have the solution as a distributed cloud solution. |
|  | Status | This option is decided. |
|  | Relationship(s) |  |
|  | Rationale of decision | This option is decided because it will be easier to adjust the required capacity to the customer base. In the unlikely event of a reduced customer base, it is possible to scale down in the cloud. |

### Performance

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Concern (Identifier: Description) | | Ensuring that alarm messages (e.g. from a fire detector) is being handled within a limited timeframe |
|  | Ranking criteria (Identifier: Name) | | 1. Performance |
|  | Options | Identifier: Name | Prioritizing certain emergency messages |
|  | Description | The Event Dispatcher shall be able to prioritize certain messages coming from the, potentially millions of, sensors. This way we can minimize the wait time for an emergency message. |
|  | Status | This option is decided. |
|  | Relationship(s) | Event Dispatcher |
|  | Evaluation | Emergency processes must in nature be top priority in a system which handles the safety of people in a Smart Home. E.g. receiving messages about a low level of milk in the fridge, must never be evaluated before a message concerning e.g. a fire. |
|  | Rationale of decision | Emergency messages are the single most important messages in the system; therefore these must be treated accordingly. |
|  | Identifier: Name | No prioritizing needed |
|  | Description | The event dispatcher simply computes messages as they come, and by guaranteeing fast computations (perhaps via a clever algorithm); we can ensure some maximum delay (worst case) off all message handling. |
|  | Status | This option is declined. |
|  | Relationship(s) | Event Dispatcher |
|  | Rationale of decision | There simply is no argument for not handling emergency messages as top priority. The system, even using some “perfect” algorithm, should therefore always choose emergency messages over non-emergency ones. |

### Security

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Concern (Identifier: Description) | | The system shall be able to authenticate users in a secure way. |
|  | Ranking criteria (Identifier: Name) | | 1. Security |
|  | Options | Identifier: Name | The User uses a Smart Phone application, in order to request access to the Smart Home |
|  | Description | When the given user arrives at his/her Smart Home, he/she is able to access a Smart Phone Application which can open up the House. This application is protected by a password, in order to be opened. This way the system knows when doors should be opened up, but also when a door is being opened without proper user authentication (hence starts a break-in process). |
|  | Status | This option is decided. |
|  | Relationship(s) | House Access Control |
|  | Evaluation | The House Access Control is a key point in the security aspect of the solution. The system must ensure that no unauthorized people get access to the house. With that being said, different approaches might be considered, but since this one seems to be the one requiring the least support/interaction from the supplier, this is the best candidate. |
|  | Rationale of decision | Other methods might also handle this requirement (see coming options), but this option separates itself from the rest, in a way that the supplier is not required to deliver additional hardware, in order to maintain a secure House Access Control. The assumption is, though, that users do have a Smart Phone, and in case it gets lost (the phone), that communication to other members of the house is possible, so that they can open up the house locks mechanisms. |
|  | Identifier: Name | Card scanner |
|  | Description | A solution where the users receive security cards (classic authentication security plastic cards) when buying the desired solution. Furthermore the given House gets a Card Reader installed, so that the users are required to swipe their card, as well as entering a e.g. 4-digit pin code. |
|  | Status | This option is declined. |
|  | Relationship(s) | House Access Control |
|  | Rationale of decision | As a supplier this complicates things a lot. This creates the need for the supplier to manufacture these security cards, when new users get attached to the system, but also when existing cards gets lost. Furthermore the physical devices that are able to scan these cards must be installed at every Smart Home, which again, potentially could be the entire world population. The difference with a card reader device, compared to all other devices in a Smart Home, is that this would then be a minimum requirement for the system. |

# References

Standards: OIC (Intel), Allseen (Qualcom)

<http://www.datacenterknowledge.com/archives/2015/07/23/the-iot-standards-war/>

Ziggbee – The wireless language

<http://www.zigbee.org/>

OMA (Open mobile Alliance)

<http://openmobilealliance.hs-sites.com/lightweight-m2m-specification-from-oma>

Device discovery Options

<http://postscapes.com/iot-device-discovery>

“DIAT: A Scalable Distributed Architecture for IoT” - by: Chayan Sarkar et. Al.

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7000513>

“IoT (Internet of Things) and DfPL (Device-free Passive Localisation) in a disaster management scenario” (Article in Simulation Modelling Practice & Theory, AUGUST 2012) – by: Gabriel Deak et. Al.

## Additional Links

<http://www.networkworld.com/article/2456421/internet-of-things/a-guide-to-the-confusing-internet-of-things-standards-world.html>

<http://electronicdesign.com/iot/open-standards-will-enable-iot-s-growth>

# Notes

Pictures from meeting:

